Multi-agent Artificial Intelligence (COMP0124)

Description

Aims:
The module is intended to provide an introduction of multi-agent machine learning, a subfield of Artificial Intelligence (AI). Multi-agent learning arises in a variety of domains where multiple intelligent computerised agents interact not only with the environment but also with each other. There are an increasing number of applications ranging from controlling a group of autonomous vehicles/drones to coordinating collaborative bots in factories and warehouses, optimising distributed sensor networks/traffic, and machine bidding in competitive e-commerce and financial markets, just to name a few. The module combines the study of machine learning with that of game theory and economics, including topics such as game theory, auction theory, algorithmic mechanism design, multi-agent (deep) reinforcement learning. Practical applications, including online advertising, online auction, adversarial training for generative models, bots planning, and AI agents playing online games, will also be covered and discussed.

Learning outcomes:
On successful completion of the module, the students will be able to master both the theoretical and practical aspects of module. Specifically, the students will understand the underlying principle and the theory for decision making by multiple parties, and the learning algorithms that obtain optimal decision or reach an equilibrium from different objectives. The students are also expected to be able to make use of the learned theory and algorithms to formulate and solve large-scale practical learning problems where multiple objectives/incentives co-exist.

Content:
Game theory and online auction:
- The prisoner's dilemma, dominant strategy, Nash equilibrium, Mixed strategies, and Pareto optimality;
- English Auctions, Dutch Auctions, the first price auctions,

Key information

Year 2019/20
Credit value 15 (150 study hours)
Delivery PGT L7, Campus-based
Reading List View on UCL website
Tutor Prof Jun Wang
Term Term 2
Timetable View on UCL website

Assessment

Report: 50%
Coursework: 50%

Find out more

For more information about the department, programmes, relevant open days and to browse other modules, visit ucl.ac.uk
and the second price auctions; Learning Nash Equilibria and Learning in Repeated Games:
- The linear programming solution and the Lemke-Howson algorithm; Single-agent Reinforcement Learning:
- Value Iterations, Policy Iterations, Q-learning, Policy Gradient, and Deep Reinforcement Learning;

Multi-agent reinforcement learning:
- Stochastic games, Nash-Q, Gradient Ascent, WOLF, and Mean-field Q learning;

Applications:
- Online advertising machine bidding, AI agents playing online games, and learning to collaborate for bots.

Reference books:

Requisites:
In order to be eligible to select this module, a student must be registered on a programme for which it is a formally-approved option or elective choice AND must have (i) strong competency in programming in Python/Java (evidence of at least one past programming project is required); (ii) strong competency in probability and statistics; and (iii) knowledge of machine learning and deep learning methods and algorithms, for example, classification, regression and clustering (evidence of at least one past programming project using TensorFlow, PyTorch, MXNet or similar deep learning frameworks is required).
Multi-agent Artificial Intelligence (COMP0124)

**Description**

**Aims:**
The module is intended to provide an introduction of multi-agent machine learning, a subfield of Artificial Intelligence (AI). Multi-agent learning arises in a variety of domains where multiple intelligent computerised agents interact not only with the environment but also with each other. There are an increasing number of applications ranging from controlling a group of autonomous vehicles/drones to coordinating collaborative bots in factories and warehouses, optimising distributed sensor networks/traffic, and machine bidding in competitive e-commerce and financial markets, just to name a few. The module combines the study of machine learning with that of game theory and economics, including topics such as game theory, auction theory, algorithmic mechanism design, multi-agent (deep) reinforcement learning. Practical applications, including online advertising, online auction, adversarial training for generative models, bots planning, and AI agents playing online games, will also be covered and discussed.

**Learning outcomes:**
On successful completion of the module, the students will be able to master both the theoretical and practical aspects of module. Specifically, the students will understand the underlying principle and the theory for decision making by multiple parties, and the learning algorithms that obtain optimal decision or reach an equilibrium from different objectives. The students are also expected to be able to make use of the learned theory and algorithms to formulate and solve large-scale practical learning problems where multiple objectives/incentives co-exist.

**Content:**
Game theory and online auction:
- The prisoner’s dilemma, dominant strategy, Nash equilibrium, Mixed strategies, and Pareto optimality;
- English Auctions, Dutch Auctions, the first price auctions,
and the second price auctions; Learning Nash Equilibria and Learning in Repeated Games:
- The linear programming solution and the Lemke-Howson algorithm; Single-agent
Reinforcement Learning:
- Value Iterations, Policy Iterations, Q-learning,
Policy Gradient, and Deep Reinforcement Learning;

Multi-agent reinforcement learning:
- Stochastic games, Nash-Q, Gradient Ascent,
WOLF, and Mean-field Q learning;

Applications:
- Online advertising machine bidding, AI agents playing online games, and learning to collaborate for bots.

Reference books:
- Shoham, Yoav, and Kevin Leyton-Brown. Multiagent systems:
  - Algorithmic, game-theoretic, and logical foundations. Cambridge University Press, 2008;

Requisites:
In order to be eligible to select this module, a student must be registered on a programme for which it is a formally-approved option or elective choice AND must have (i) strong competency in programming in Python/Java (evidence of at least one past programming project is required); (ii) strong competency in probability and statistics; and (iii) knowledge of machine learning and deep learning methods and algorithms, for example, classification, regression and clustering (evidence of at least one past programming project using TensorFlow, PyTorch, MXNet or similar deep learning frameworks is required).